

LOAN DOCUMENT

		PHOTOGRAPH THIS SHEET												
DTIC ACCESSION NUMBER		LEVEL		INVENTORY										
	DOCUMENT IDENTIFICATION													
		DISTRIBUTION STATEMENT												
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2" style="text-align: center;">ACCESSION TO</td> </tr> <tr> <td style="width: 50%;">NTIS</td> <td style="width: 50%;">GRAM</td> </tr> <tr> <td>DTIC</td> <td>TRAC</td> </tr> <tr> <td>UNANNOUNCED</td> <td></td> </tr> <tr> <td>JUSTIFICATION</td> <td></td> </tr> </table>		ACCESSION TO		NTIS	GRAM	DTIC	TRAC	UNANNOUNCED		JUSTIFICATION		<div style="border: 1px solid black; height: 150px; width: 100%;"></div>		
ACCESSION TO														
NTIS	GRAM													
DTIC	TRAC													
UNANNOUNCED														
JUSTIFICATION														
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2" style="text-align: center;">BY</td> </tr> <tr> <td colspan="2" style="text-align: center;">DISTRIBUTION/</td> </tr> <tr> <td colspan="2" style="text-align: center;">AVAILABILITY CODES</td> </tr> <tr> <td style="width: 50%; text-align: center;">DISTRIBUTION</td> <td style="width: 50%; text-align: center;">AVAILABILITY AND/OR SPECIAL</td> </tr> <tr> <td style="height: 50px; vertical-align: bottom; font-size: 2em;">A-1</td> <td></td> </tr> </table>		BY		DISTRIBUTION/		AVAILABILITY CODES		DISTRIBUTION	AVAILABILITY AND/OR SPECIAL	A-1		DATE ACCESSIONED		
BY														
DISTRIBUTION/														
AVAILABILITY CODES														
DISTRIBUTION	AVAILABILITY AND/OR SPECIAL													
A-1														
DISTRIBUTION STAMP														
		DATE RETURNED												
DATE RECEIVED IN DTIC		REGISTERED OR CERTIFIED NUMBER												
<div style="border: 1px solid black; padding: 10px; font-size: 2em; margin: 10px auto; width: 80%;">19981223 048</div>														
PHOTOGRAPH THIS SHEET AND RETURN TO DTIC-FDAC														

HANDLE WITH CARE

ADDC

Recd. Info.

REED-SOLOMON ENCODER/DECODER

COMPUTER PROGRAM PERFORMANCE SPECIFICATION

FEBRUARY, 1976

8000746

REED-SOLOMON ENCODER/DECODER
COMPUTER PROGRAM PERFORMANCE SPECIFICATION

FEBRUARY, 1976

Prepared under Contract N62269-75-C-0503

DATA ITEM A004

for

Department of the Navy
Naval Air Development Center

by

ITT Avionics Division
500 Washington Avenue
Nutley, N. J. 07110

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

DD FORM 1473
1 JAN 73

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT

ITT Avionics has built and successfully tested a RSED laboratory breadboard that was funded under contract N62269-75-C-0503 (Naval Air Development Center). A summary of the engineering tests is listed below.

Encoding Time	<150 microseconds
Round Trip Timing Detection	< 20 microseconds
Decode Time	Decode time is dependent on Errata. Refer to section 5 of report number D11801 for decode times.

TABLE OF CONTENTS

		PAGE NO.
1.0	Scope	1
1.1	Identification	1
1.2	Introduction	1
1.2.1	Encoder	1
1.2.2	Decoder	1
1.2.3	Microprocessor Functions	1
2.0	References	2
3.0	Requirements	3
3.1	Functional Description	3
3.1.1	Microprocessor Interface Block Diagram	3
3.1.2	Equiment Descriptions	3
3.1.3	Microprocessor Input/Output	9
3.1.4	Function Description	15
APPENDIX A	Reed-Solomon Software Algorithm Specification Number RS-10	

FIGURES

FIGURE	PAGE NO.
3.1 RSED Block Diagram	4
3.2 Microprocessor Interface Block Diagram	5

TABLES

TABLE	PAGE NO.
3.1 Microprocessor Read/Write Control	7
3.2 RSED Service Priorities	8
3.3 Microprocessor Input/Output	11

1.0 SCOPE

1.1 Identification

The Reed - Solomon Encoder/Decoder (RSED) Microprocessor Program is identified as RSED1.

1.2 Introduction

The Reed - Solomon linear block codes of N ($N \leq 31$) symbols and information rate K/N ($K < N$) are maximum distance separable codes with symbols from a finite field of 32 elements. The selected standard code with 31 symbols and information rate 15/31 can correct sixteen symbol erasures (E) or eight symbol errors (e) or any combination of errors and erasures where $2e + E \leq 16$.

1.2.1 Encoder

The error correction coding process is a (31, 15) Reed - Solomon code. The code to be used is generated by the generating function $G(X)$, where coefficients are elements of the Galois field of 2^5 equals 32. If α is the root of the primitive polynomial

$$X^5 + X^2 + 1 = 0$$

then the generating function of the Reed - Solomon code is given by

$$G(X) = \prod_{i=1}^{16} (X + \alpha^i)$$

Also, a short code (16,4) is used which is a shortened* (31,15) codeword.

1.2.2 Decoder

The decoding of the Reed - Solomon code words will be accomplished as described by Berlekamp (1,3) with modification suggested by Lin (2). Reference section 2.

The major steps required to decode the received code words are:

- o Syndrome Computation
- o Erasure Location Computation
- o Modified Syndrome Computation
- o Error Location Computation
- o Error and Erasure Value Computation
- o Errata Correction

1.2.3 Microprocessor Functions

The encoding and syndrome computation functions are performed by special RSED nonmicroprocessor hardware. The remaining functions listed in section 1.2.2 are performed by the microprocessor.

* Truncated (last four characters) version of the (20,4) which is a subset of the (31,15) e.g. (31-L, 15-L) where $L = 11$.

2.0 REFERENCES

- o Berlekamp, Elwyn R. ,
Algebraic Coding Theory
New York: McGraw Hill, 1968
- o Lin, Shu
An Introduction to Error Correcting Codes
Englewood Cliffs, New Jersey: Prentice Hall, 1970
- o Berlekamp, Elwyn R. , ed.
Key Papers in the development of Coding Theory
New York: IEEE Press, 1974
- o Peterson, W. W. and Weldon, E. J. , Jr.
Error Correcting Codes, 2nd Ed.
Cambridge, Mass: MIT Press, 1972
- o ITT Avionics
Final Engineering Report, Reed-Solomon Encoder/Decoder,
Data Item A003, Contract N62269-75-C-0503, NADC, 1975
- o 3001 Microprogram Control Unit
Technical Memo Number MCS-268-0275/27.5K
INTEL Corporation, Santa Clara, California
- o 3002 Central Processing Element
Technical Memo Number MCS-269-0275/27.5K
INTEL Corporation, Santa Clara, California

3.0 REQUIREMENTS

3.1 Functional Description

3.1.1 Microprocessor Interface Block Diagram

Figure 3.1 is the RSED Block Diagram which shows the relationship of the microprocessor to the other RSED equipment.

RSED interface signals as defined by ICWG agreements, reference ITT Avionic's RSED Final Report Data Item A003, are routed through the Input and Output Interface Blocks.

Encoder functions and RTT detection functions are performed simultaneously with and independently of microprocessor functions.

Decoding functions are shared between the syndrome generator and the microprocessor.

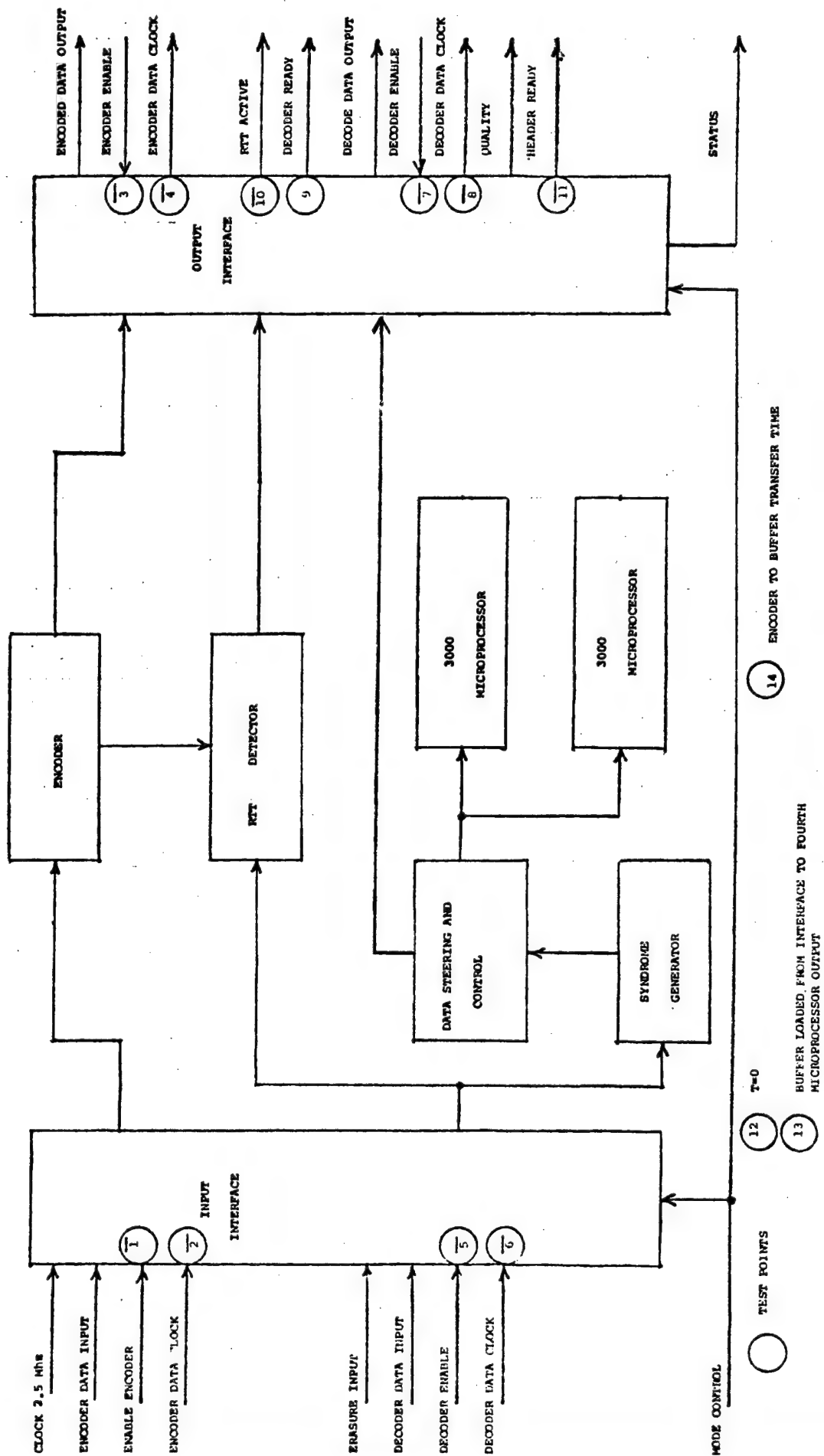
Figure 3.2 is the Microprocessor Interface Block Diagram.

There are two microprocessors which work in parallel independently of each other. Input/Output control flags are set by the Data Steering and Control Unit (DSCU) and by the microprocessors in the shared Data Rams. Data is passed between the DSCU and the microprocessors by depositing it in the appropriate Data Ram.

3.1.2 Equipment Descriptions

The DSCU is the RSED unit interfacing with the microprocessors. The functions of the DSCU with respect to the microprocessors are as follows:

- o To initialize the microprocessor after power has been turned on. Initialization forces the microprocessor to begin execution of the microprogram at the instruction placed in position Row 0 Column 15 in the ROM. (Reference the INTEL document 3001 Microprogram Control Unit) This is the master reset function.
- o To reinitialize the microprocessors at the beginning of each slot. Requirements for processing are that the RSED should accept and process one header word and three data words every slot period. The duration of a slot is 7.8125 milliseconds. The DSCU performs a time monitoring function over the microprocessors by providing a reinitialization pulse at the start of each slot to terminate processing if the microprocessors fail to complete their tasks within the required time constraints. This is the cycle reset function.



RSED BLOCK DIAGRAM

FIGURE 3.1

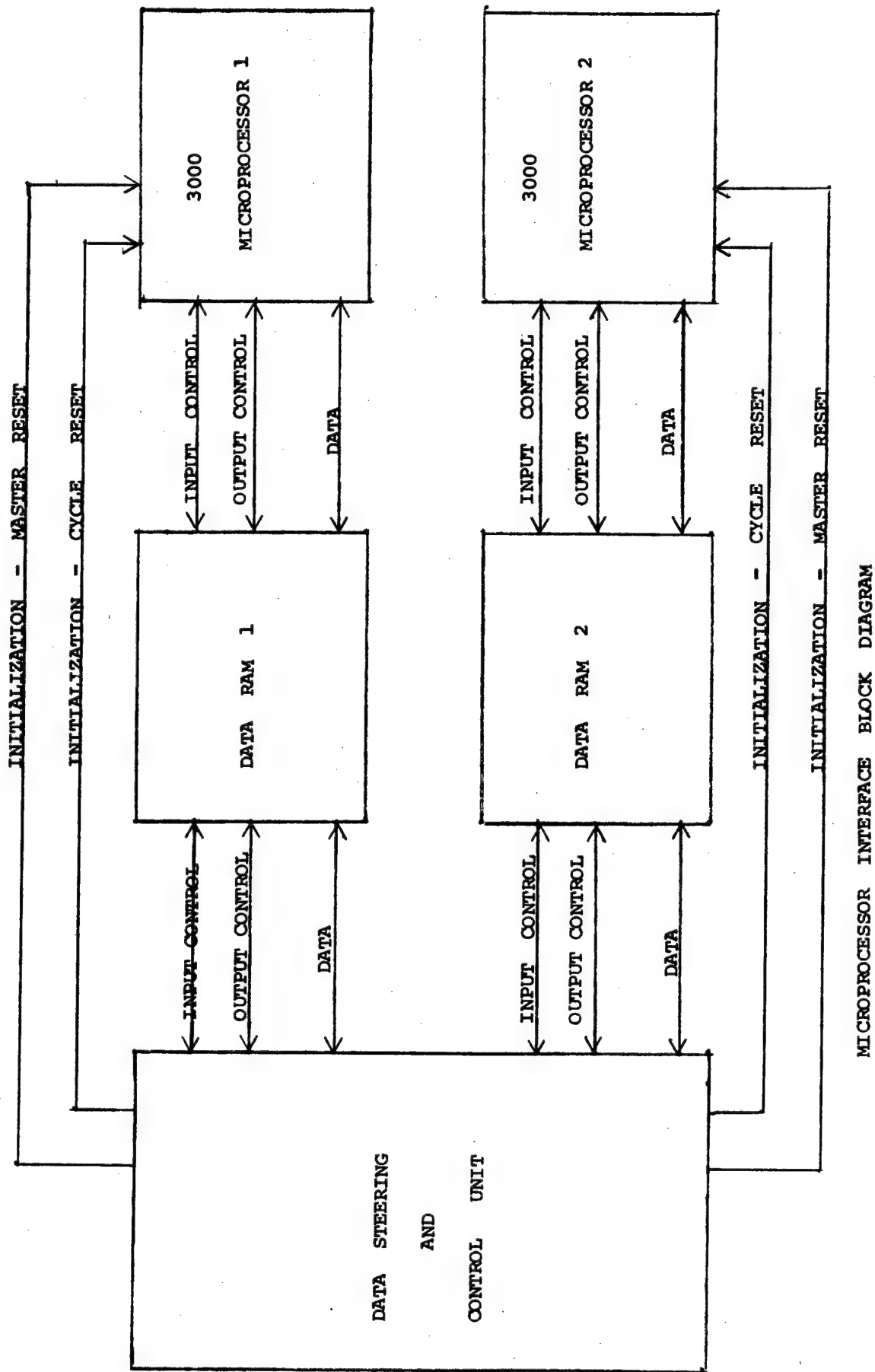


FIGURE 3.2

- o The transfer of data from the DSCU to the microprocessors for processing and the return of the processed data is achieved by the setting and resetting of data control flags stored in the appropriate Data Rams. The necessary control requirements observed by both the DSCU and the microprocessor are shown in Table 3.1 .

Write requests, i.e. a request for the transfer of unprocessed data from the DSCU to the microprocessor, are made by the microprocessor setting the ZY flag to the value 01 . The recognition of this request and the positive response of data transfer into the RAM by the DSCU is communicated to the microprocessor by the DSCU setting the ZY flag to the value 11 on completion of the transfer. Until this setting is made the microprocessor cannot process data. As soon as the microprocessor begins to process the data it resets the ZY flag to the value 00 .

When the microprocessor has finished processing the data it makes a read request to the DSCU by setting the X flag to the value 1 . The DSCU will not read data from the RAM until the X flag is set to the value 1 . When the DSCU has completed the transfer of processed data out of the RAM it resets the X flag to the value 0 .

Two areas are reserved in each Data Ram for the data transferred. Each area is both an input and an output area. The R/S code word that is placed in input area A will be read out of the same area A after the results of processing have been written appropriately into area A. Each microprocessor will process only one word at a time but will set the request flag for a second word before the processing of the first word is completed. This permits the transfer of the second word into the Data Ram to be completed or at least started before the microprocessor is ready to begin to process the next word. The second input/output area is referred to as area B.

Access to the Data Rams is shared by the DSCU and the corresponding microprocessor. See Figure 3.2 . The microprocessor has the first level of access priority and exercises this on a per instruction basis. If a Read or Write request is inserted into a microinstruction, access to the Data Ram is given to the microprocessor when the instruction is executed. When the microinstruction does not include a Read or Write request, Data Ram access is given to the DSCU. The transfer of a block of data to or from the Data Ram may therefore be interrupted in a time shared memory access manner.

The DSCU is the control unit for all RSED hardware units. It exercises control by cycling through the set of priority levels shown in Table 3.2 . Reference Figure 3.1 for RSED units which require a service priority. Table 3.2 shows the relationship between the microprocessors and other RSED units with respect to service priority. The priorities are organized to correspond to external system requirements, encoder time constraints and to decode data flow with respect to the syndrome generator and the microprocessors which share the decode function.

MICROPROCESSOR READ/WRITE CONTROL

<u>FLAG</u>	<u>MEANING</u>	<u>SET BY:</u>
X = 0	Data not ready	DSCU (on read completion)
X = 1	Data ready	Microprocessor (on decode completion)
ZY = 00	No write request made	Microprocessor (when processing begins)
ZY = 01	Write request made (DSCU may write)	Microprocessor (when processing of a word is near completion)
ZY = 11	DSCU has filled input buffer	DSCU (on write completion)
ZY = 10	Not used	

NOTES

1. A set of Read/Write control flags exist for each input/output area in a microprocessor
2. The flags are stored in fixed locations in the Data Ram used by each microprocessor as follows:

RAM ADDRESS (decimal)

126	X Flag	Area A
127	zY Flag	Area A
254	X Flag	Area B
255	ZY Flag	Area B

TABLE 3.1

RSED SERVICE PRIORITIES

<u>LEVEL</u>	<u>EQUIPMENT/FUNCTION</u>	<u>CONDITION</u>
1	External System Data Input/Output (Time synchronized event externally controlled)	Always
2	Microprocessor Data Ram Read Out	Always
3	Encoder Data Read Out	Always
4	Encoder Data Read In	Always
5	Syndrome Generator Data Read Out	Always
6	Microprocessor Data Ram Write In	If Syndrome Generator has data ready for read out
7	Syndrome Generator Data Write In	Always

TABLE 3.2

When either a microprocessor Data Ram Read out or Write in service priority is reached in the service cycle, the DSCU searches for a Read/Write request in the following order:

1. microprocessor 1, area A
2. microprocessor 1, area B
3. microprocessor 2, area A
4. microprocessor 2, area B
5. microprocessor 3, area A
6. microprocessor 3, area B

Control has been provided for three microprocessors with an implementation of two.

If a service request is found in the service cycle, the corresponding function is initiated. After completion of the function, the service cycle is restarted from priority level two unless a priority level one occurs. A priority level one is a time synchronized event controlled by the external system and is immediately honoured when it occurs. At this time the microprocessors receive a cycle reset which reinitializes them.

Transfer of data will occur between only one microprocessor and the DSCU at any point in time. Data transfer is also only in one direction at a time. Both these statements are true during the transfer of the whole block of data.

3.1.3 Microprocessor Input/Output

The input and output data sent to and from the microprocessor under the control of the DSCU is shown in Table 3.3 . The word numbers used in this table correspond to Data Ram addresses (decimal).

As described in section 3.1.2 four (4) R/S code words, one a header word (short word) and three data words (long words) are sent to the RSED every slot time. The DSCU sends these words to either of the two microprocessors in the order three long words followed by the short word in the fourth position.

A long R/S code word, (31/15), consists of fifteen (15) data characters and sixteen (16) parity characters. A short R/S code word, (16/4) consists of four (4) data characters and twelve (12) parity characters. The same number of characters are transferred to the microprocessor for each type of word, the missing characters in a short word being filled out with eleven (11) leading zeros and four (4) trailing erasures. A block of eighty-four (84) eight (8) bit words are transferred to the Data Ram by a data write function. Refer to words with address four (4) thro eighty-seven (87) in Table 3.3 . The block consists of the following data types:

- o Addresses 4 thru 7 contain label words.
The only field used by the microprocessor on input is the 1 bit code select field in word 4. It is used to interpret the input data, the value 0 indicating a long word and the value 1 indicating a short word.
- o Addresses 8 thru 22 contain data characters.
Each data character or leading zero is written into one Data Ram word.
- o Addresses 23 thru 38 contain parity characters.
Each parity character or erasure fill out character is written into one Data Ram word.
- o Addresses 40 thru 55 contain syndromes.
The syndromes are the output of the syndrome generator. There are always sixteen syndromes stored one per Data Ram word.
- o Address 56 contains the number of erasures.
The syndrome generator computes the number of erasures found. The number is sent to the Data Ram for use by the microprocessor.
- o Addresses 57 thru 87 contain erasure position numbers.
A block of 31 words is always sent to the Data Ram. The maximum number of erasures in any R/S word is thirty one. The number of erasures stored in address 56 is used to interpret the validity of this group of words.

A block of thirty-five (35) eight (8) bit words are transferred out of the Data Ram by a data read function. Only nineteen (19) of these words are sent out to the external system as called for by ICWG agreements, reference ITT Avionic's RSED Final Report Data Item A003. The block consists of the following data types:

- o Addresses 4 thru 7 contain label words.
On output data, the microprocessor writes into bits 0 thro 4 of word 7, the data quality word. Bit 4 is set to zero if the R/S word was successfully decoded and bits 0 thro 3 contain the number of errors found. Bit 4 is set to a one if the R/S word could not be decoded, bits 0 thro 3 do not have meaning then.
- o Addresses 8 thru 22 contain data characters.
On output data, these are corrected data characters as applicable if the R/S word was successfully decoded. If the word was not decoded the characters are those sent into the Data Ram.
- o Addresses 23 thru 38 contain parity characters.
These characters are left uncorrected by the microprocessor.

MICROPROCESSOR INPUT/OUTPUT

RAM ADDRESS	BIT POSITION						DESCRIPTION	
	* 5	4	3	2	1	0		
0 1 2 3							UNALLOCATED	
4		Word # Bit 0	Block # MSB	Block # LSB	0 Long 1 Short	Code Select	Label	
5		Word # Bit 5	Word # Bit 4	Word # Bit 3	Word # Bit 2	Word # Bit 1	Label	
6		Debug Flag	Debug Flag	Word # Bit 8	Word # Bit 7	Word # Bit 6	Label	
7		Errata MSB	Errata	Errata	Errata	Errata LSB	Label	
8		MSB				LSB	31/15 C30	16/4 "0"s
9		MSB				LSB	C29	"0"s
10		MSB				LSB	C28	"0"s
11		MSB				LSB	C27	"0"s
12		MSB				LSB	C26	"0"s
13		MSB				LSB	C25	"0"s
14		MSB				LSB	C24	"0"s
15		MSB				LSB	C23	"0"s
16		MSB				LSB	C22	"0"s
17		MSB				LSB	C21	"0"s
18		MSB				LSB	C20	"0"s
19		MSB				LSB	C19	} DATA
20		MSB				LSB	C18	
21		MSB				LSB	C17	
22		MSB				LSB	C16	
23		MSB		11		LSB	C15	Parity

* See last chart page.

Table 3.3 (Continued) Page 2 of 4

RAM ADDRESS	BIT POSITION						DESCRIPTION	
	*5	4	3	2	1	0		
24		MSB				LSB	31/15 C14	16/4
25		MSB				LSB	C13	
26		MSB				LSB	C12	
27		MSB				LSB	C11	
28		MSB				LSB	C10	
29		MSB				LSB	C09	PARITY
30		MSB				LSB	C08	
31		MSB				LSB	C07	
32		MSB				LSB	C06	
33		MSB				LSB	C05	
34		MSB				LSB	C04	
35		MSB				LSB	C03	
36		MSB				LSB	C02	4 Erasures
37		MSB				LSB	C01	
38		MSB				LSB	C00	
39		MSB				LSB	SPARE	
40		MSB				LSB	S1	
41		MSB				LSB	S2	
42		MSB				LSB	S3	Syndromes
43		MSB				LSB	S4	
44		MSB				LSB	S5	
45		MSB		12		LSB	S6	

*See last chart page.

Table 3.3 (Continued) Page 3 of 4

RAM ADDRESS	BIT POSITION						DESCRIPTION
	*5	4	3	2	1	0	
46		MSB				LSB	S7
47		MSB				LSB	S8
48							S9
49							S10
50							Syndromes
51							S11
52							S12
53							S13
54							S14
55							S15
56							S16
57		MSB				LSB	Number of Erasures
58							31/15 16/4
59							E_L
60							1. There are as many E_L 's as there are erasures.
61							2. The E_L 's are packed
62							3. The E_L 's are in descending order 30 thro 0
63							4. The maximum number of E_L 's is 31.
64							5. Positions are given as binary numbers
65							
66							E10
67							E11
68							E12

*See last chart page.

RAM ADDRESS	BIT POSITION						DESCRIPTION
	* 5	4	3	2	1	0	
69							E13
70							E14
71							E15
72							E16
73							E17
74							E18
75							E19
76							E20
77							E21
78							E22
79							E23
80							E24
81							E25
82							E26
83							E27
84							E28
85							E29
86							E30
87							E31

* Bit 5 is used as an Erasure Indicator which is valid only for words 8 thro 38.

3.1.4 Function Description

The functions to be performed by the microprocessor program are part of the process of decoding a R/S code word. Erasure position location, the computation of the number of erasures and the syndrome generation are performed by non microprocessor RSED hardware units. Reference ITT Avionic's Final Report Data Item A003, Appendix A, Reed-Solomon Hardware Functional Specification, Number RS-9, for a description of these functions. The results of these computations are transferred to the Data Ram for use by the microprocessor which completes the decoding process. The functions performed by the microprocessor program are as follows:

o Erasure Polynomial Generation

The erasure polynomial is defined as the following product

$$\prod_{i=1}^s (1 + \beta_{E_i} X) \quad \begin{array}{l} \text{for all } i = 1, \dots, s \\ \text{where } s = \text{number of erasures} \\ \text{and } E_i = \text{erasure locations} \\ \text{in power form} \end{array}$$

The microprocessor program computes this product and stores the result in the Data Ram.

o Modified Syndrome Computation

The modified syndromes are defined as a set of terms T_j derived from the product of the syndromes and the coefficients of the erasure polynomial in the following manner:

$$T_j = \sum_{i=0}^s S_{s+j-i} \sigma_{E_i} \quad \begin{array}{l} \text{where } S_i = \text{Syndromes} \\ \quad \quad i = 1, 2, \dots, 16 \\ \text{and } \sigma_{E_i} = \text{erasure polynomial} \\ \quad \quad \quad \text{coefficients} \\ \quad \quad i = 0, 1, \dots, s \\ \quad \quad s = \text{number of erasures} \\ \\ \text{and } j = 1, 2, \dots, (16-s) \end{array}$$

The microprocessor program computes these terms and stores the results in the Data Ram.

o Error Location Polynomial Generation

The error polynomial is defined as:

$$\sigma^{(u)}(x) = \sum_{i=0}^n \sigma_i^{(u)} x^i$$

where

u = current iteration, u = -1, 0, 1,(16-s)

s = number of erasures

The generation of the error polynomial is an iterative process, successive iterations depending on the results of previous iterations. The (u+1)th iteration is defined as:

Equation 1

$$\sigma^{(u+1)}(x) = \sigma^{(u)}(x) + du d\epsilon^{-1} x^{u-\epsilon} \sigma^{(\epsilon)}(x)$$

where

ϵ = a selected previous iteration

du = a coefficient index, GF32 element

lu = a decimal computation number

u-lu = a decimal computation number

Initial iterations are defined as follows:

Iteration u	Polynomial $\sigma^{(u)}(x)$	du	lu	u-lu
-1	1	1	0	-1
0	1	T_1	0	0

where

T_m = the modified syndromes

and

m = 1, 2,(16-s)

Using these initial iterations, succeeding iterations can be computed using Equation 1 (defined above) and Equations 2 and 3 (defined below) .

Equation 2

$$\begin{aligned} du &= \sum_{\forall m+n = u+1} T_m \sigma_n^{(u)} \\ &= T_{u+1} \sigma_0^{(u)} + \dots + T_0 \sigma_{u+1}^{(u)} \end{aligned}$$

where
 σ s are the coefficients of $\sigma^{(u)}(X)$

and

if $du = 0$, then $\sigma_{u+1}^{(u)}(X) = \sigma_u^{(u)}(X)$ and $l_{u+1} = l_u$

Equation 3

When $du \neq 0$, p is a previous iteration selected so that

$$dp \neq 0$$

$$l_{u+1} = \text{maximum of } l_u \text{ and } l_p + (u - p)$$

The microprocessor program computes the coefficients of the error polynomial and stores the results in the Data Ram.

0 Solving for Roots of the Error Location Polynomial

The error location polynomial is defined as

$$\sigma_e(X) = \sum_{n=0}^t \sigma_{e_n}^{(X)} X^n$$

If the roots of the polynomial are defined as

$$\alpha^j, \text{ where } j = 0, 1, \dots, t$$

then

$$\sum_{n=0}^t \sigma_{e_n}(\alpha^j) = 0 \quad \text{for all values of } j.$$

The microprocessor program determines which values of α cause the polynomial expression to reduce to zero. It then determines the inverse values of the α values. The inverse values are defined as β_{e_i} where

$$\beta_{e_i} = \alpha^{31-j} \quad \text{when } \sigma_e(\alpha^j) = 0$$

The program stores the β values in the Data Ram. These inverse root values are the locations of the errors in the R/S code word.

o Errata Polynomial Generation

The errata (i.e. the combination of errors and erasures) polynomial is a function of the syndromes, the erasure polynomial and of the error polynomial. It is defined as $\tilde{Z}(X)$ where $Z(X)$ is defined as

$$Z(X) = (1 + S(X)) \sigma_E(X) \sigma_e(X)$$

γ_i are the coefficients of $Z(X)$

$$\text{and } \tilde{Z}(X) = X^{s+t} Z(1/X)$$

$$= \sum_{i=s+t-A}^{s+t} \gamma_{s+t-i} X^i \quad \text{where } A \leq s+t$$

Only the coefficients $X^0, X^1, X^2, \dots, X^{s+t}$ are required.
and

$S(X)$ = The syndromes S_1, S_2, \dots, S_{16}

$\sigma_E(X)$ = The coefficients of the erasure polynomial

$\sigma_e(X)$ = The coefficients of the error polynomial

The microprogram computes the required coefficients and stores the results in the Data Ram.

o Errata Value Calculation

An errata value is defined as the value which must be added to an erroneous received character in the R/S word being decoded. Addition in this process is Galois addition, which is equivalent to an exclusive or function. Received data characters which are determined to be either erasures or errors will be corrected. The received parity characters will not be corrected.

If β_{E_i} and β_{e_i} are the erasure and error locations, then

the errata locations are defined as

$$\beta_i, \text{ for } i = (E_i, e_i)$$

The errata correction value C_i is defined as

$$C_i = \frac{\tilde{Z}(\beta_i)}{\prod_{i \neq j} (\beta_i + \beta_j)} \quad \text{for all } i, j = E_i \text{ or } e_i$$

and

$$C_i = 0 \text{ (00000)} \quad \text{for all } i, j \neq E_i \text{ or } e_i$$

$\tilde{Z}(X) = \tilde{Z}(\beta_i)$ is as defined in the Errata Polynomial Generation Function.

The microprogram computes the required errata correction values and corrects the corresponding received R/S character which is stored in the appropriate input/output area as described in sections 3.1.2 and 3.1.3.

o Evaluation of R/S Code Words Without Errata Function

The microprocessor program will detect the absence of errata in the R/S code word. The absence of erasures is determined by the value of the ' number of erasures ' field, input to the microprocessor. The absence of errors is determined by an examination of the syndromes which are also input to the microprocessor. If all the syndromes each have the value of zero, then there are no errors in the R/S code word.

o Decode Failure Functions

The microprocessor program will perform the following decode failure checks:

1. If the number of erasures is greater than 16, a decode failure will be declared.
2. If the following relationship is true, a decode failure will be declared.

$$2e + E > 16 \quad \text{where} \quad e = \text{the number of errors}$$

$$E = \text{the number of erasures}$$

3. If the number of distinct (nonequal) roots found is less than the degree of the error polynomial, then a decode failure will be declared.

The occurrence of any one of these conditions is an indication that the power of the code has been exceeded.

o Data Quality Function

The microprocessor program will provide an indication of the quality of the data received. When an R/S code word has been decoded an indication of this fact is inserted into an output label word, together with the number of errors (not erasures) detected. When an R/S code word has been declared a decode failure, then an indication of this failure is provided. Reference section 3.1.3 for details of the data quality word, word 7 of the input/output area.

APPENDIX A

REED-SOLOMON SOFTWARE ALGORITHM

SPECIFICATION NUMBER RS-10

TABLE OF CONTENTS

			<u>PAGE NO.</u>
1.0 Erasure Polynomial Generator	$\sigma_E(X)$	APGEN	1
2.0 Modified Syndromes	T_j	MSYNG	2
3.0 Error Location Polynomial	$\sigma_e(X)$	EPGEN	4
4.0 Error Locations Inverse Roots of $\sigma_e(X)^{-1}$	β_{e1}	EPVAL	10
5.0 Modified Errata Evaluator Polynomial	$\tilde{Z}(X)$	RPGEN	11
6.0 Errata Value Calculation	C'_1	RPVAL	13

1.0 Erasure Polynomial Generator (APGEN)

1.1 Input

$E_1, 1 = 1, \dots, s$ Erasure Location Numbers

(If $r_K \equiv$ erasure, $E_1 = K, \forall K$)

1.2 Output: $s+1$ coef. $\sigma_{E_i} \quad i = 0, \dots, s$

$\sigma_{E_0} \equiv 1$, Coef. of Erasure Polynomial

$$\begin{aligned}\sigma_E(X) &= \sum_{i=0}^s \sigma_{E_i} X^i = \prod_{l=1}^s (1 + \alpha^{E_l} X) \\ &= \prod_{l=1}^s (1 + \beta_{E_l} X) ;\end{aligned}$$

where $\beta_{E_1} =$ Inverse roots of $\sigma_E(X) = \alpha^{E_1}$

1.3 Process:

In general

$$\sigma_{E_0} = 1$$

$$\sigma_{E_1} = \sum_{E_1} \beta_{E_1}$$

$$\sigma_{E_2} = \sum_{E_1, E_2, i \neq j} \beta_{E_i} \beta_{E_j}$$

$$\sigma_{E_{s-1}} = \sum_{E_j} \left(\prod_{i=1}^s \beta_{E_i} / \beta_{E_j} \right)$$

Simplify notation $\beta_i \equiv \beta_{E_i}$

can compute σ_{E_i} by the following iterative process.

Define $A_i(s) = \sigma_{E_i}$ $i = 0, \dots, s$

where $A_i(j)$ = the j^{th} value of A_i iterate till $j = s$

$$A_i(j) = A_{i-1}(j-1) \beta_j + A_i(j-1) ; i < j$$

$$A_i(i) = A_{i-1}(i-1) \beta_i + 0 ; j=i$$

$$A_i(j) \equiv 0 ; i > j$$

$$A_0(j) = A_0(j-1) \equiv 1 ; \forall j$$

First iteration $j = 1$

$$A_0(1) = 1$$

$$A_1(1) = \beta_1$$

$j = 2$

$$A_0(2) = 1$$

$$A_1(2) = A_0(1) \beta_2 + A_1(1) = \beta_2 + \beta_1$$

$$A_2(2) = A_1(1) \beta_2 + A_2(1) = \beta_1 \beta_2 \text{ etc.}$$

2.0 Modified Syndromes T_j MSYNG

2.1 Inputs

$$S_i \quad i = 1, \dots, 16$$

$$\sigma_{E_i} \quad i = 0, \dots, s \quad \sigma_{E_0} \equiv 1$$

Simplify notation $\sigma_{E_i} = \sigma_i$

2.2 Output

$$T_j \quad j = 1, \dots, (16-s)$$

2.3 Process:

$$T_j = \sum_{i=0}^j S_{s+j-i} \sigma_i$$

i.e., $T_1 = S_{s+1} + \sigma_1 S_s + \sigma_2 S_{s-1} + \dots + \sigma_s S_1$

$$T_2 = S_{s+2} + \sigma_1 S_{s+1} + \sigma_2 S_s + \dots + \sigma_s S_2$$

$$T_{16-s} = S_{16} + \sigma_1 S_{15} + \sigma_2 S_{14} + \dots + \sigma_s S_{16-s}$$

Array: $\sigma \quad y(1) \quad y(2)$

X_0	σ_0	$Y_0(1)$	S_{s+1}	$Y_0(2)$	S_{s+2}	Push down Stack
X_1	σ_1	$Y_1(1)$	S_s	$Y_1(2)$	S_{s+1}	
X_2	σ_2	$Y_2(1)$				
\vdots	\vdots	\vdots	\vdots	\vdots		
X_s	σ_s	$Y_s(1)$	S_1	$Y_s(2)$	S_2	

$$T_1 = \sum_{i=0}^s X_i Y_i(1)$$

$$T_2 = \sum_{i=0}^s X_i Y_i(2)$$

2.4

Modified Syndromes Berlekamp Notation

In general case, define T_j^1 as the coefficient of X^j in the equation.

$$(1 + S(X)) \sigma_E(X) = T^1(X) = \sum_{j=0}^{s+16} T_j^1 X^j$$

$$S(X) = \sum_{i=1}^{16} S_i X^i$$

$$\sigma_E(X) = \sum_{i=0}^s \sigma_{E_i} X^i, \quad s = \# \text{ of erasures}$$

Then T_i 's on previous page are $T_i = T_{s+i}^1$

3.0 Error Location Polynomial $\sigma_e(X)$ EPGEN

3.1 Input

Modified Syndromes T_j^1 (5 bits)

$$j = 1, \dots, (16-s)$$

Where s = number of Erasures

If $s = 0$; $T_j = S_j$, $j = 1, \dots, 16$

3.2 Output

$t + 1$ Coefficient of Error Location

Polynomial (5 bits) σ_{e_i} $i = 0, \dots, t$

t = number of errors

where $\sigma_{e_0} \equiv 1$

3.3 Process - Simplify notation $\sigma_{e_i} = \sigma_i$

3.3.1 Option 1 Berlekamp Algorithm

This is iterative Process $K = 0, \dots, (16-s)$

Define values of K^{th} iteration (depends only on $K-1$)

$$\sigma^{(K)}(X) = \sum_{i=0}^n \sigma_i^{(K)} X^i = \text{Error Loc. Polynomial}$$

$\sigma_i^{(K)}$ GF (32) element

$$\tau^{(K)}(X) = \sum_{i=0}^m \tau_i^{(K)}(X) X^i = \text{Tau Polynomial}$$

$\tau_i^{(K)}$ = GF(32) element

$B(K)$ = Computation Number (Decimal)

$D(K)$ = Computation Number (Decimal)

$\Delta^{(K)}$ = Coefficient index - GF(32) element

0th Iteration

Define $\sigma^{(0)}(X) \equiv 1$

$\tau^{(0)}(X) \equiv 1$

$$D(o) \equiv 0$$

$$B(o) \equiv 0$$

In general

$$\Delta^{(K)} = \sum_{m+n=k+1} T_m \sigma_n^{(K)} = T_{K+1} \sigma_o^{(K)} + T_K \sigma_1^{(K)} + \dots + T_o \sigma_{K+1}^{(K)}$$

Define $T_o \equiv 1$

i.e., $\Delta^{(K)} =$ Coefficient of X^{K+1} in $(1+T(X)) \sigma^{(K)}(X)$

$$T(X) = 1 + \sum_{i=1}^{16-A} T_i X^i$$

$$\sigma^{(K+1)}(X) = \sigma^{(K)}(X) + \Delta^{(K)} \tau^{(K)}(X) X$$

$$\text{or } \sigma_i^{(K+1)} = \sigma_i^{(K)} + \Delta^{(K)} \tau_{i-1}^{(K)} \quad \forall i \leq \text{max degree} \\ (\sigma^{(K)}(X), \tau^{(K)}(X))$$

The value of $D(K+1)$, $B(K+1)$ and $\tau^{(K+1)}(X)$ are computed from the two equations A and B by the rules below.

$$A: D(K+1) = D(K)$$

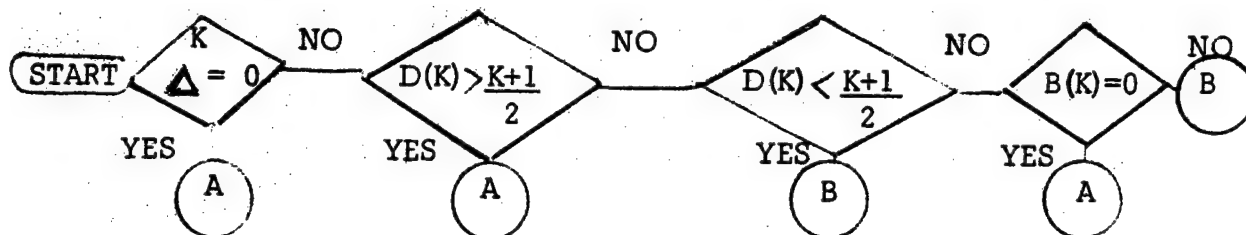
$$B(K+1) = B(K)$$

$$\tau^{(K+1)}(X) = \tau^{(K)}(X) \text{ or } \tau_i^{(K+1)} = \tau_i^{(K)}$$

B: $D(K+1) = K+1-D(K)$ (decimal arithmetic)

$B(K+1) = 1-B(K)$ (decimal arithmetic)

$$\tau_i^{(K+1)} = \sigma_i^{(K)} \frac{1}{\Delta^{(K)}}$$



Continue until $K = 16-s$ and $\bar{\sigma}_e(X) = \bar{\sigma}^{(16-s)}(X)$.

3.3.2

Option 2 Shu Lin Algorithm (Berlekamp Modified)

Iterative process with reliance on earlier results,

u^{th} iteration depends on $u-1$, $u-2$, etc..

Define values of u^{th} iteration, $u = -1, \dots, (16-s)$.

$$\sigma^{(u)}(X) = \sum_{i=0}^n \sigma_i^{(u)} X^i = \text{error location polynomial}$$

d_u = Coef Index GF32 element $\equiv \Delta^{(K)}$

l_u = Computation number.(decimal)

$u-1_u$ = Computation Number. (decimal)

th
-1 iteration

th
0 iteration

Define:

$$\sigma^{(-1)}(X) = 1$$

$$\sigma^{(0)}(X) = 1$$

$$d_{-1} = 1$$

$$d_{0 \ 1} = T$$

$$l_{-1} = 0$$

$$l_0 = 0$$

$$\sigma^{(16-s)}(X) \equiv \sigma_e(X)$$

In general, compute $u+1$ st row according to value of

d_u by (A') or (B') where:

$$d_u = \sum_{\forall m+n=u+1} T_{m \ n} \sigma^{(u)} = T_{u+1 \ 0} \sigma_{0}^{(u)} + \dots + T_{0 \ u+1} \sigma_{u+1}^{(u)}$$

$$T_{0 \ 0} = 1$$

A': If $d_u = 0$ (00000)

$$\sigma_{u+1}(X) = \sigma^{(u)}(X)$$

$$l_{u+1} = l_u$$

B': If $d_u \neq 0$ find previous row, ρ , such that $d_{\rho} \neq 0$

$$\text{and } (\rho - l_\rho) \geq (\tau - l_\tau) \quad \forall \tau < u$$

(or $(\rho - l_\rho)$ is maximum)

$$\text{Then } \tau^{(u+1)}(x) = \tau^{(u)}(x) + d_u d_\rho^{-1} x^{u-\rho} \tau^{(\rho)}(x)$$

$$\text{or } \tau_i^{(u+1)} = \tau_i^{(u)} + d_u d_\rho^{-1} \tau_{i-(u-\rho)}^{(\rho)}$$

$$l_{u+1} = \text{Max} \left[l_u, l_{\rho+u-\rho} \right]$$

4.0 Inverse Roots of $\sigma_e(X) = \beta_{e_i} \quad i = 1, \dots, t$ EPVAL

4.1 Input

$t + 1$ coefficients of $\sigma_e(X) = \sigma_{e_i} \quad i = 0, \dots, t$

$$\sigma_{e_0} \equiv 1$$

4.2 Output

t distinct inverse roots of $\sigma_e(X)$

If there are less than t roots then decoding is in error and an error will be flagged.

$$\beta_{e_i} \quad i = 1, \dots, t$$

$$\beta_{e_i} = \alpha^{e_i} \quad \text{error in location} \quad r_{e_i}$$

4.3 Process

Compute $\sigma_e(X) \quad X = \alpha^j$

If $\sigma_e(\alpha^j) \equiv \phi, (00000)$

Then $\beta_{e_i}^{-1} = \alpha^j$ and $\beta_{e_i} = \alpha^{31-j}$

$$\sigma_e(\alpha^j) = \sum_{n=0}^t \sigma_{e_n} (\alpha^j)^n$$

can be computed sequentially as S_K

$$\sigma_e(\alpha^j) = ((\sigma_{e_t} \alpha^j + \sigma_{e_{t-1}}) \alpha^j + \sigma_{e_{t-2}}) \alpha^j + \dots$$

5.0 Errata Evaluator Polynomial $\tilde{Z}(X)$ RPGEN

5.1 Input

Sixteen Syndromes $S_1 \dots S_{16}$

S_1

s = coefficients of the Erasure Polynomial

σ_{E_0}

σ_{E_s}

σ_{E_1}

t = coefficients of the Error Location

σ_{e_0}

$\dots \sigma_{e_t}$

σ_{e_1}

Polynomial

5.2 Output

Coefficients of $\tilde{\gamma}_1$ (a variable number)

5.3 Process

$$Z(X) = (1 + S(X)) \sigma_E(X) \sigma_e(X) = \sum_{i=0}^{s+t} \gamma_i X^i$$

$$= (1 + S(X)) \sigma_E(X) \sigma_e(X) \text{ mod } Z^{s+t+1}$$

Only the coefficients of $X^0, X^1, X^2, \dots, X^{s+t}$ are required

where s = # of erasures = the degree of $\sigma_E(X)$

t = # of errors = the degree of $\sigma_e(X)$

$$\therefore \gamma_i = \sum \sigma_{E_1} \sigma_{e_m} S_K$$

$$\gamma_0 \equiv 1, \sigma_{e_0} \equiv 1, \sigma_{E_0} \equiv 1$$

for all $K + 1 + m = i$

Note:

It is possible for $\gamma_i = 0$ for $i > A$, $A < s+t$

$$\therefore Z(X) = \sum_{i=0}^A \gamma_i X^i \quad A \leq s+t$$

$$\text{Now } \tilde{Z}(X) = X^{s+t} Z(1/X) = \sum_{i=s+t-A}^{s+t} \tilde{\gamma}_i X^i$$

$$\text{where } \tilde{\gamma}_i = \gamma_{s+t-i} \quad i = (s+t-A), \dots, (s+t)$$

6.0 Errata Value Calculator (RPVAL)

6.1 Input

s+t values β_{E_i}, β_{e_i}

all inverse roots of $\sigma_E(X)$ and $\sigma_e(X)$

where $\beta_{E_i} = \alpha^{E_i}$
 $\beta_{e_i} = \alpha^{e_i}$ } Denotes errata in received
character r_{E_i} and r_{e_i}

Let β_{E_i} and β_{e_i} be denoted by β_i

where $i = E_i$ or e_i

Then $\beta_i = \alpha^i$ $i = (E_i, e_i) = \text{errata location}$

6.2 Output

C'_i = Errata value for location i

$i = 16, \dots, 30$ (all data locations)

6.3 Process

$C'_i = 0$ (00000) $\forall i \neq E_i$ or e_i

$C'_i = \frac{\tilde{Z}(\beta_i)}{\beta_i \prod_{i \neq j} (\beta_i + \beta_j)}$ $\forall i, j = E_i$ or e_i

$$\tilde{z}(\beta_i) = \sum_{n=s+t-A}^{s+t} \tilde{\gamma}_n (\beta_i)^n$$

can be computed sequentially as in S_k computation

$$\tilde{z}(\beta_i) = ((\tilde{\gamma}_{s+t}\beta_i + \tilde{\gamma}_{s+t-1})\beta_i + \tilde{\gamma}_{s+t-2})\beta_i \dots$$

Option 2:

$$C'_i = \frac{z(\beta_i)^{-1}}{\prod_{i \neq j} (1 + \beta_j / \beta_i)^{-1}}$$

$$\text{where } \beta_i^{-1} = \beta_{e_i}^{-1} = \alpha^{31-e_i}$$

$$\beta_i^{-1} = \beta_{E_i}^{-1} = \alpha^{31-E_i}$$